# PROPERTIES OF A HAND-PRINTED CHINESE CHARACTER RECOGNIZER BASED ON CONTEXTUAL VECTOR QUANTIZATION

*Ping-Chong CHEE, Sau-Lai LEUNG, Pak-Kwong WONG and Chorkin CHAN*

Department of Computer Science
The University of Hong Kong
Pokfulam Road, Hong Kong

## ABSTRACT

A hand-printed Chinese character recognizer based on Contextual Vector Quantization (CVQ) has been built previously. In this paper, several properties of the recognizer will be discussed and the recognizer of 4516 Chinese characters has a successful rate of 91.0%. Then the output of the recognizer is passed to a language model which when applied to recognize a passage of about 1200 characters raises the rate from 91.5% to 97.5%.

## 1. INTRODUCTION

Previously, an off-line hand-printed Chinese character recognizer based on Contextual Vector Quantization (CVQ) [1] has been built. In CVQ, a feature vector derived from each pixel of the image is quantized to a codeword by considering not just the vector itself but its neighbors and their codeword identities as well. In this paper, we are going to discuss several properties of the recognizer and the effect of using a language model in the postprocessing phase.

## 2. CVQ MODELING OF CHARACTERS

A character image is represented by a matrix of feature vectors $\mathbf{O} = [\mathbf{o}_{i,j}]$ with $\mathbf{o}_{i,j}$ observed at pixel $(i,j)$. Each "observed datum" $\mathbf{o}_{i,j}$, to be modeled as a realization of a random vector, can be interpreted as a partly observed version of a "complete datum" $\mathbf{y}_{i,j} = (\mathbf{o}_{i,j}, z_{i,j})$, where $z_{i,j}$ is the missing value (hidden state) and takes one of the $K$ qualitative values $\mathcal{G} = \{G_1, G_2, \cdots, G_K\}$ to represent a particular region of a Chinese character. The "missing data" image $\mathbf{Z} = [z_{i,j}]$ represents the corresponding machine states when these regions are generated.

CVQ labeling of an image is to quantize the pixels individually on the basis of their posterior probabilities of state membership given all the observed feature vectors of the image, i.e., pixel $(i,j)$ is quantized on the basis of maximizing the posterior probability $\Pr(z_{i,j}|\mathbf{O})$. In order to reduce the complexity of the problem, $z_{i,j}$ is chosen to maximize $\Pr(z_{i,j}|\mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}})$, where $\eta_{i,j}$ is the neighborhood of pixel $(i,j)$. Under the assumption that feature vectors in the same neighborhood are related to each other through their states only and $z_{m,n}$'s, where $(m,n) \in \eta_{i,j}$, are mutually independent except its dependency on $z_{i,j}$, then given a character image with observation feature vectors $[\mathbf{o}_{i,j}]$, each $\mathbf{o}_{i,j}$ is assigned to state $G_k$ if

$$G_k = \underset{z_{i,j}}{\operatorname{argmax}} \Pr(z_{i,j}) \cdot f(\mathbf{o}_{i,j}|z_{i,j}) \cdot \prod_{(m,n)\in\eta_{i,j}} \sum_{z_{m,n}} \Pr(z_{m,n}|z_{i,j}) \cdot f(\mathbf{o}_{m,n}|z_{m,n}) \qquad (1)$$

where the terms in the second line of (1) represent the contribution of contextual information.

A CVQ model for each character can be generated by finding the corresponding $\Pr(z_{i,j})$'s, $f(\mathbf{o}_{i,j}|z_{i,j})$'s and $\Pr(z_{m,n}|z_{i,j})$'s from the training samples. Let there be a collection of $C$ such models, $\Lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_C\}$, where $\lambda_c$ denotes the set of parameters of the $c$-th model. Given an unknown character image $\mathbf{O}$, the corresponding missing state image $\mathbf{Z}^{(c)} = [z_{i,j}^{(c)}]$ is identified with the CVQ labeling method using each CVQ model in turn. Then, with the discriminant function defined for class $\omega_c$ as one of those in (3) to (7), the image $\mathbf{O}$ will be classified to $\omega_d$ if

$$g(\mathbf{O};\lambda_d) > g(\mathbf{O};\lambda_c) \quad \forall c \neq d \qquad (2)$$

## 3. PROPERTIES OF THE CVQ RECOGNIZER

To investigate various properties of the recognizer, a vocabulary of 470 characters corresponding to 5 sections of the GB 2312-80 set is used instead of using the whole vocabulary.

0-7803-3192-3/96 $5.00©1996 IEEE

$$g_1(\mathbf{O};\lambda_c) = \prod_{(i,j)} f(\mathbf{o}_{i,j}|z_{i,j}^{(c)}) \tag{3}$$

$$g_2(\mathbf{O};\lambda_c) = \prod_{(i,j)} f(\mathbf{o}_{i,j}|z_{i,j}^{(c)}) \cdot \prod_{(m,n)\in\eta_{i,j}} \Pr(z_{m,n}^{(c)}|z_{i,j}^{(c)}) \tag{4}$$

$$g_3(\mathbf{O};\lambda_c) = \prod_{(i,j)} \Pr(z_{i,j}^{(c)}) \cdot f(\mathbf{o}_{i,j}|z_{i,j}^{(c)}) \cdot \prod_{(m,n)\in\eta_{i,j}} \Pr(z_{m,n}^{(c)}|z_{i,j}^{(c)}) \tag{5}$$

$$g_4(\mathbf{O};\lambda_c) = \prod_{(i,j)} \Pr(z_{i,j}^{(c)}) \cdot \prod_{(m,n)\in\eta_{i,j}} f(\mathbf{o}_{m,n}|z_{m,n}^{(c)}) \cdot \Pr(z_{m,n}^{(c)}|z_{i,j}^{(c)}) \tag{6}$$

$$g_5(\mathbf{O};\lambda_c) = \prod_{(i,j)} \Pr(z_{i,j}^{(c)}) \cdot f(\mathbf{o}_{i,j}|z_{i,j}^{(c)}) \cdot \prod_{(m,n)\in\eta_{i,j}} \sum_{z_{m,n}^{(c)}} \Pr(z_{m,n}^{(c)}|z_{i,j}^{(c)}) \cdot f(\mathbf{o}_{m,n}|z_{m,n}^{(c)}) \tag{7}$$

### 3.1. Discriminant Functions

By making different assumptions, different discriminant functions are examined.

$g_1$ in (3) is equivalent to the discriminant function adopted for Dynamic Programming in speech recognition.

$g_2$ in (4) is equivalent to the discriminant function used for the Viterbi path using Hidden Markov Model (HMM) for speech recognition.

$g_3$ in (5) is the same as $g_2$ with $\Pr(z_{i,j})$ added.

$g_4$ in (6) is a generalization of $g_3$.

$g_5$ in (7) is the discriminant function based on CVQ.

The system performance for these discriminant functions are shown below:

| Dis. Func. | $g_1$ | $g_2$ | $g_3$ | $g_4$ | $g_5$ |
|---|---|---|---|---|---|
| recog. rate (%) | 83.2 | 94.2 | 92.3 | 97.0 | 97.0 |
| recog. time (s) | 22.4 | 25.2 | 25.2 | 24.3 | 16.9 |

Table 1: Performance of various discriminant functions

$g_5$ needs much less time than the remaining four because the terms in $g_5$ have been calculated during the state assignments of the observation feature vectors in (1). Since both the recognition rate and time favor $g_5$, it is chosen for all later experiments.

### 3.2. I/O Time

It can be observed that the time required to recognize a character is quite long considering that all experiments are performed on an IBM RS/6000 3BT workstation. The main reason is due to the time required for bringing in codebooks from disk. The average size of a codebook is about 50K bytes. In order to recognize an unknown character, all the codebooks corresponding to the 470 characters which amounts to about 23.5M bytes of data have to be read. From the total time required to recognize a character, about 65% of the time is used for I/O, which is a very high percentage of overhead.

To solve the problem, the recognition process is done in batch mode. The codebooks are read once to recognize for every, say, 400 unknown characters, so that the overhead can be shared. The time required to recognize an unknown is thus reduced from 16.9s to 5.7s.

### 3.3. Preclassification

To further reduce the recognition time, preclassification is done to reduce the number of possible identities of an unknown for the CVQ recognizer.

A global feature vector of seven dimensions are constructed which consists of the population of each of the connectivity number (CN) [5] and the percentage of each pixel type. These features are chosen because they are simple to calculate, and they reflect the structure of a character.

The connectivity number is calculated for each black pixel of a character image. Then the number of branching elements (whose CN = 0, 3 or 4), edge elements (whose CN = 1) and connecting elements (whose CN = 2) are found.

In pixel type identification, each black pixel is classified to either a part of a horizontal stroke, vertical stroke, south-west slanting stroke or south-east slanting stroke. The lengths of the contiguous path extended from a pixel are found in all the four directions. The pixel is classified to the direction which gives the longest length. For example, to find the length of the contiguous path in the horizontal direction for pixel $(i,j)$, the pixels adjacent are traced one by one until there is no more contiguous black pixel or $|j-y| \geq 2$ where $y$ is the y-coordinate of a pixel traced. The num-

ber of traced pixels is the length of stroke in the horizontal direction. The other three directions are done in similar ways. The percentages of the four types of strokes are then computed to serve as the remaining four dimensions of the feature vector.

Each training sample is divided into 4 quadrants. In each quadrant, the mean and the variance of each feature are calculated. For a given unknown, the features are found and the Euclidean distances with each of the 4516 characters are computed with negligible time. The $N$ codebooks corresponding to the smallest distances are passed to the CVQ recognizer for final recognition. The rate of including the correct codebook against $N$ is summarized below:

| $N$ | 300 | 400 | 500 | 600 |
|---|---|---|---|---|
| preclass. rate (%) | 98.7 | 99.0 | 99.3 | 99.5 |

Table 2: Preclassification rate against no. of candidates

## 4. EXPERIMENTAL DETAILS

112 samples of each of the 4516 most commonly used simplified characters [2] are collected from 112 writers. Among them, 102 are used for training and the rest for testing. The characters are scanned by a 300dpi scanner into Tagged Image File Format (TIFF) files, which are then noise removed [3], thinned [4], segmented and size normalized.

Since there are roughly the same number of testing samples for each of the 4516 character, the recognition rate is weighted by the usage frequency [2] of each character. With the above setup, a recognition rate of 91.0% and an average recognition time of 2.6s are achieved.

## 5. LANGUAGE MODEL

If the input is a sequence of words, its linguistic information can provide another useful basis for correct recognition of the input characters [6]. The second phase of the character recognizer is a language model which endows the recognizer with the lexical knowledge in Chinese. For each input character, the language model chooses the best one out of the ten best character candidates generated by the first phase of the recognizer based on its lexical knowledge.

The language model is supported by a lexicon in which there are over 85,000 Chinese words, and each word has an entry consisting of its GB code and usage frequency. The vocabulary of the lexicon covers nearly all the Chinese words which can be found in general Chinese texts. A large Chinese text corpus of over 63,000,000 characters has been used to train the lexicon to increase its vocabulary and adjust the word usage frequencies.

For reducing the time spent in searching a word in the lexicon, an efficient word searching algorithm which is much faster than hash searching has been designed and implemented. The number of characters in a word in the lexicon varies from one to ten. In this algorithm, words having more than two characters are divided into sub-word fragments which have either one or two characters. According to their lengths, these elements are grouped into two sub-lexicons in which the index of an element is a part of its GB code. Using this part of the GB code of a word as the bin number, Bin Search Algorithm can be applied to search the word in the lexicon. The procedure of searching a word having more than two characters can be divided into a number of steps. In each of them, a fragment having one or two characters is searched for. Because most words used in Chinese texts are either one-character or two-character words, such a lexicon construction can provide a very high efficiency in word searching.

For each input character, the output of the first phase is ten character candidates, each associated with a score to measure its similarity to the input character. Since each unknown character has ten candidates, there is a lot of possible combinations to form a sequence of words. The time spent in exhaustive evaluation of all these combinations is unacceptably high. Because the first phase of the recognizer provides a high enough recognition rate, the starting point of the language model should be the sequence of the best candidates of the input characters. First, a lexical analysis algorithm called "Maximum Matching Based on Usage Frequency" is applied to segment this character sequence into words. The algorithm performs Maximum Matching to segment the character sequence first, and then word frequencies are used to overcome the ambiguity of word segmentation. In the word sequence generated by the lexical analyzer, the possibility of correct segmentation is evaluated based on the word lengths, the usage frequencies of the words, and the likelihood scores of the corresponding characters. Only those suspected words are considered to be replaced by others.

A passage of about 1200 characters is used to test the language model. The output from the first phase of these 1200 characters is 91.5% and the language model can increase the recognition rate to 97.5%. Most of the time spent in applying the language model is to load the lexicon whose size is less than 1 Mbytes into memory.

## 6. CONCLUSION

A CVQ based recognizer of handprinted Chinese characters supporting a vocabulary of 4516 characters has been proven to be successful. A simple maximum matching language model of word usage frequencies applied to this recognizer can improve its accuracy by 6.0%.

## 7. REFERENCES


[1] S.-L. Leung, P.-C. Chee, C. Chan and Q. Huo, "Contextual Vector Quantization Modeling of Hand-printed Chinese Character Recognition," *Proc. IEEE ICIP-95*, Vol. 3, pp. 432-435, 1995.

[2] *Xiandai Hanyu Pinlu Cidian, Beijing Institute of Linguistics*, pp. 1300-1387, 1986.

[3] P.-C. Chee and C. Chan, "Random Noise Removal of Binary Text Images Using Seed Filling," *Proc. IASTED SIP-95*, pp. 421-423, 1995.

[4] S. Suzuki and K. Abe, "Sequential Thinning of Binary Pictures Using Distance Transformation," *Proc. IEEE ICPR'86*, pp. 289-292, 1986.

[5] S. Yokoi, J.I. Toriwaki and T.Fukumura, "An Analysis of Topological Properties of Digitized Binary Pictures Using Local Features," *Computer Graphics and Image Processing*, Vol. 4, pp. 63-73, 1975.

[6] K.T. Lua, "From Character to Word – An Application of Information Theory," *Computer Processing of Chinese and Oriental Languages*, Vol. 4, No. 4, pp. 304-313, March 1990.